

SCFHLA: Un Modelo de Interoperabilidad Semántica para Simulación Distribuida de Cadenas de Suministro

Juan Leonardo Sarli¹, Horacio Leone¹, Milagros Gutierrez²

{juanleonardosarli, hleone}@santafe-conicet.gob.ar, mmgutier@frsf.utn.edu.ar

¹ INGAR Conicet - UTN, Avellaneda 3657, 3000, Santa Fe, Argentina.

² CIDISI UTN, Lavaisse 610, 3000, Santa Fe, Argentina.

DOI: [10.17013/risti.30.34-50](https://doi.org/10.17013/risti.30.34-50)

Resumen: La simulación distribuida de cadenas de suministro tiene la gran ventaja de preservar la independencia de los miembros de la cadena, pudiendo reutilizar simuladores existentes sin necesidad de crear uno nuevo. Sin embargo, el problema que emerge en este tipo de simulación es la necesidad de acordar el conjunto de objetos, eventos, interacciones y métricas, que deben ser entendidas por todos los participantes para lograr con éxito un resultado valioso para los mismos. En este trabajo se presenta un marco conceptual basado en una red de ontologías, que da soporte a las tareas de modelado y composición de la simulación distribuida de cadenas de suministro para garantizar la interoperabilidad semántica de sus miembros. Se utiliza el estándar HLA (High Level Architecture) como herramienta de construcción de una simulación distribuida.

Palabras-clave: Interoperabilidad Semántica; Red de Ontologías; Cadena de Suministro; Simulación Distribuida; HLA.

SCFHLA: A Semantic Interoperability Model for Supply Chain Distributed Simulation

Abstract: Supply chain distributed simulation has as major advantage to maintain members' autonomy, being able to reuse existing simulators and avoiding making a new one. However, a new problem emerges. It is required an agreement about the meaning of objects, events, interactions and metrics, to be shared during simulation in order to successfully achieve the simulation objectives. In this work, we present a conceptual framework based on an ontology network, which is the frame to define the modeling and simulation of supply chain in a distributed environment, guaranteeing semantic interoperability among members. The HLA (High Level Architecture) standard has been used as a tool to build distributed simulation.

Keywords: *Semantic Interoperability; Ontology Network; Supply Chain; Distributed Simulation; HLA.*

1. Introducción

La cadena de suministro (CS) es una estructura organizacional adecuada para que las empresas puedan mantenerse ágiles y competitivas. En estas estructuras, para mantener la administración eficiente de los flujos de información, materiales y económicos, se hace necesario fortalecer las relaciones de comunicación, colaboración, intercambio de información y confianza entre sus miembros. En las últimas décadas, las CS han traspasado los límites de las organizaciones, dando como resultado organizaciones colaborativas y en red (Cigolini, Pero, Rossi, & Sianesi, 2014; Mustafee, Katsaliaki, & Taylor, 2014). En este contexto, la construcción de un modelo de la CS es importante para comprender sus relaciones, procesos, interacciones y roles entre otros. El modelo de referencia SCOR (Supply Chain Council, 2012) es uno de los más utilizados para modelar CS. Este modelo brinda herramientas de diagnóstico y evaluación comparativa, que ayudan a las organizaciones a detectar oportunidades de mejora. El modelo SCOR presenta una visión estática de CS, si además al mismo se le añade una perspectiva dinámica se contaría con información más precisa sobre la cual tomar decisiones.

Las técnicas de simulación, emergen como una herramienta poderosa para encontrar configuraciones deseables que alcanzan los objetivos propuestos de CS. Además, mejoran la toma de decisión conjunta, basada en información detallada y confiable. La configuración más utilizada para una simulación de CS es un único modelo simple ejecutado en una única computadora. Esta configuración no permite reusar simuladores existentes de los miembros de la CS e implica un trabajo costoso en cuanto a tiempo, ejecución y recopilación de información, dada la naturaleza distribuida y la alta independencia entre los miembros de la CS (Anagnostou & Taylor, 2017; Mustafee, Taylor, Katsaliaki, & Brailsford, 2009). Sin embargo esta configuración evita la composición de modelos y la transferencia de datos entre sus miembros (Mustafee, Taylor, Katsaliaki, Dwivedi, & Williams, 2012). En este contexto, la simulación distribuida (SD) aparece como una configuración más apropiada para ejecutar una simulación de CS dado que refleja de mejor manera la naturaleza propia de la misma. Además, la SD permite el reuso de simuladores existentes en los miembros de la CS, manteniendo así la independencia y evitando la necesidad de construir un único simulador que comprenda el comportamiento de todos sus participantes. Cada miembro preserva su lógica de negocio, utiliza su modelo de simulación y comparte la mínima cantidad de información necesaria, lo que permite modificar su lógica interna sin afectar al resto (Anagnostou & Taylor, 2017).

El estándar High Level Architecture (HLA) (IEEE, 2010a) es el más utilizado para desarrollar SD. El uso del mismo garantiza la interoperabilidad sintáctica a través de la definición de la interfaz y del documento FOM (Federation Object Model) que representa el contrato bajo el cual los federados participarán de la Federación. El FOM provee las etiquetas necesarias para interpretar correctamente interacciones y objetos, sin embargo los valores que tomen estas etiquetas deben ser semánticamente acordados para que la simulación tenga resultados válidos (Tolk, Bair, & Diallo, 2013; Tolk, Diallo, & Padilla, 2012). Este acuerdo se conoce como interoperabilidad semántica, la cual se considera una solución para el tratamiento del significado de la información mediante la construcción de recursos individuales semánticamente consistentes (Mezgar & Rauschecker, 2014; Piedra et al., 2015; Piedra & Suárez, 2018). El objetivo

de la interoperabilidad semántica es lograr que sistemas de información autónomos comprendan el significado de la información generada y compartida por otros sistemas. Cada miembro de una CS utiliza términos diferentes para describir conceptos similares, o términos idénticos para significar conceptos diferentes, lo que introduce confusión y errores de uso. Existen soluciones tecnológicas para la interoperabilidad semántica, las que abarcan desde vocabularios controlados, tesauros y ontologías, hasta la utilización de metadatos y estándares (Béhé, Galland, Gaud, Nicolle, & Koukam, 2014). En este trabajo, se propone el uso de una red de ontologías como la solución más adecuada. Esta red llamada *SCFHLA* (Supply Chain Federation HLA), presenta un marco teórico que da soporte a la interoperabilidad semántica entre simuladores en una SD de CS. La red conceptualiza los dominios de CS y de federaciones HLA. El objetivo de usar una red de ontologías en lugar de una única ontología es el de permitir el desarrollo modular de las mismas, facilitar su gestión y mantenimiento. De esta manera, el marco teórico provee: (i) un vocabulario común para definir el modelo de CS a simular. (ii) reglas de integridad que guían al modelador en la construcción del modelo. (iii) reglas de mapeo entre conceptos utilizados en la definición del modelo de CS a simular y conceptos necesarios de SD, y (iv) reglas de transformación para generar en forma automática el modelo de objetos (FOM).

El resto del trabajo se encuentra organizado de la siguiente forma. En la sección 2, se definen los conceptos utilizados en el desarrollo de este trabajo: HLA, ontologías y red de ontologías. En la sección 3, se presenta la red de ontologías, cada una de las ontologías que forman la red y las reglas de mapeo usadas para establecer semejanzas entre los modelos. En la sección 4, se presentan los trabajos relacionados. Luego, en la sección 5 se presenta un ejemplo donde se muestra el uso de la red en la definición de la CS, la generación automática de la federación y del FOM. Finalmente, se presentan las conclusiones y trabajos futuros.

2. Contexto

Se presenta en esta sección los conceptos teóricos utilizado en el desarrollo de este trabajo. Así, para el desarrollo de una SD se utilizó el estándar HLA. También se utilizó el concepto de red de ontologías que permitió el trabajo colaborativo, modular e iterativo del marco conceptual que se presenta.

2.1. Estándar HLA

HLA es una arquitectura estándar que soporta el reúso de capacidades disponibles en diferentes simuladores, a la vez que posibilita el desarrollo de sistemas de simulación complejos de manera cooperativa y distribuida, soportando el desarrollo de simulación basada en componentes (IEEE, 2010b). En este sentido, HLA propone integrar sistemas de simulación diferentes cada uno con un objetivo determinado, en un sistema mayor que represente el comportamiento del mismo, sin necesidad de reescribir algún componente o comenzar a crear desde cero el modelo de mayor nivel y su simulador. Un simulador que cumple las especificaciones HLA, para ser integrado en la arquitectura, se lo llama federado. Los federados luego se unen en federaciones. HLA utiliza el término federación para hacer referencia al simulador del sistema complejo, formado por simuladores de menor nivel llamados federados los cuales integran la federación.

La Figura 1 muestra los componentes de HLA. Las reglas son un conjunto de ítems que definen las responsabilidades y relaciones entre los componentes de una federación HLA. La especificación de la interfaz define la interfaz funcional entre federados y la infraestructura de ejecución (runtime infrastructure o RTI) de HLA (IEEE, 2010a). El RTI puede ser visto como el sistema operativo distribuido que provee comunicación y coordinación entre los federados, es una implementación de la especificación de la interfaz y brinda un conjunto de servicios para manejar la simulación de HLA. Este software está fuera del alcance de la especificación HLA. Existen algunas implementaciones disponibles tales como PoRTIco (“Portico,” 2016) y PITCH RTI (“Pitch Technologies,” 2017). El patrón del Modelo de Objeto (OMT Object Model Template) provee un formato común de presentación para el modelo de objetos HLA. Establece el formato de modelos claves como ser: FOM, Simulation Object Model (SOM), y Management Object Model (MOM). Particularmente, el FOM tiene como objetivo proveer una especificación común para el intercambio de datos entre federados en un formato estándar (IEEE, 2010c). En síntesis el FOM establece un contrato del modelo de información que es necesario (pero no suficiente) para alcanzar interoperabilidad entre federados. Cuando los federados se unen a la federación, se asume que éstos conocen y aceptan dicho contrato.

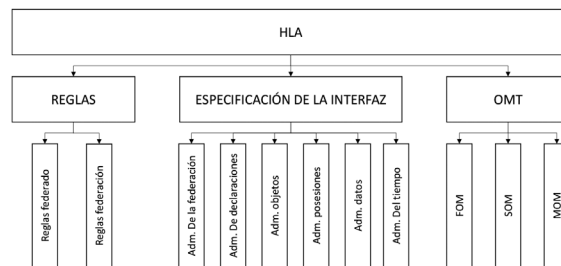


Figura 1 – Estándar HLA

2.2. Ontologías y Red de Ontologías

Una ontología es una especificación formal y explícita de una conceptualización compartida de un dominio (Hofmann, Pali, & Mihelcic, 2011). Las ontologías son apropiadas para definir explícitamente qué se entiende que existe en el dominio (Staab & Studer, 2009) y cómo cada componente se relaciona con los demás componentes (Zeigler & Hammonds, 2007). Las ontologías organizan la representación del conocimiento y capturan la información sobre objetos en un dominio de conocimiento (Gómez-Pérez, Fernandez-Lopez, & Corcho, 2010; Tolaba, Caliusco, & Galli, 2014). Una ontología es una 6-tupla de conceptos, relaciones, jerarquías, una función que relaciona conceptos de manera no taxonómica, un conjunto de axiomas y un conjunto de reglas, lo que formalmente puede expresarse de la siguiente manera (Maedche, 2002):

$$O = \{C, R, H, rel, A, DR\} \text{ donde:}$$

C y R son dos conjuntos disyuntos, C : conceptos que representan clases de objetos y R relaciones entre conceptos. H es una jerarquía de conceptos, $H \subseteq C \times C$ la cual se denomina

jerarquía de conceptos o taxonomía. La función $rel: R \rightarrow C \times C$ que relaciona los conceptos de manera no taxonómica. A es un conjunto de axiomas que expresan las propiedades del modelo. DR es un conjunto de reglas de derivación expresadas como cláusula de HORN. El lenguaje *Ontology Web Language (OWL)* es el estándar usado para implementar una ontología (W3C, 2015b). Dado que OWL no permite expresar reglas lógicas como cláusulas de HORN, es combinado con el lenguaje *Semantic Web Rule Language (SWRL)* (W3C, 2004) para especificar este tipo de reglas con conceptos de OWL.

Una red de ontologías es un conjunto de ontologías vinculadas a través de relaciones de mapeo, modularización, versión y dependencia (Allocca, D’Aquin, & Motta, 2009). En esta red, las meta-relaciones entre las ontologías son especificadas explícitamente (Diaz, Motz, & Rohrer, 2011). Cada ontología conceptualiza un dominio independiente. La principal ventaja de usar una red de ontologías es la posibilidad de desarrollar un dominio dado de una forma modular y colaborativa, donde cada ontología es más fácil de entender y mantener. Así los diseñadores pueden estar trabajando sobre las ontologías en forma concurrente.

3. La Red de Ontologías SCFHLA

La red de ontologías SCFHLA describe el contexto de SD de CS. La Figura 2 muestra la red SCFHLA, donde se identifican las dos áreas principales tenidas en cuenta y las meta-relaciones entre las ontologías. La ontología *HLAFed* conceptualiza el contexto de SD. Así los conceptos *Federation*, *Federate*, *objectModel* hacen referencia a los elementos Federación, Federado y modelo de objetos respectivamente, necesarios para construir una SD con el estándar HLA. Por otro lado, la ontología *SCK* especifica los conceptos tales como *SupplyChain*, *Relation*, *Participant*, *Metric* correspondientes a CS, relaciones, participante, métrica, indispensables para definir una CS. Las meta-relaciones *performs* y *representA* relacionan conceptos de ambas ontologías. A continuación se describen cada una de las ontologías, sus relaciones y el uso que se propone de la red.

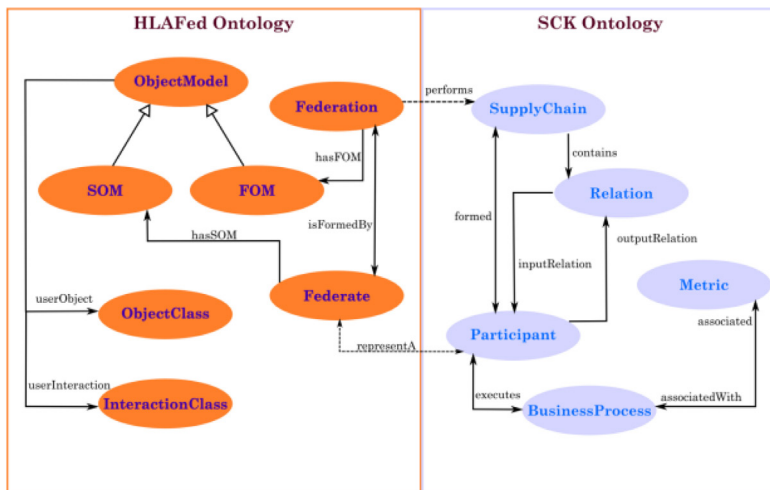


Figura 2 – Red de ontologías SCFHLA

3.1. Ontología SCK

La Figura 3 muestra en detalle los conceptos que conforman la ontología SCK. Una CS es modelada a través del concepto *SupplyChain*. Una organización que es miembro de una cadena es modelada con el concepto *Participant* y las relaciones que existen entre los miembros se modelan con el concepto *Relation*. Estas relaciones implican que existe un intercambio de información y/o materiales entre los participantes. Se consideraron relaciones de entradas (*inputRelation*) las cuales indican que el participante recibe interacciones de otro participante y relaciones de salida (*outputRelation*) que implican que un participante envía información/materiales a otro miembro. Estas relaciones tienen un conjunto de parámetros asociados que la describen, lo cual se modela a través de la relación *hasParameter* entre los conceptos *Relation* y *RelationParameter*. A su vez, los miembros pueden jugar distintos roles dentro de la cadena. Algunos roles se definieron de acuerdo a SCOR: *Make*, *Delivery*, *Source* y el rol *Authority* tiene un significado especial: identifica el participante encargado de construir el modelo válido para la simulación de la cadena.

El rol *Source* se define como aquel componente que recibe requerimientos de materias primas y/o servicios, y entrega materia prima y/o servicios. El rol *Make* es aquel que requiere y recibe materia prima pero a la vez, la transforma en un producto final. Este rol está asociado generalmente con las fábricas. El rol *Deliver* es aquel que entrega y recibe productos o servicios intermedios o finales. Generalmente se asocia con el transportista o procesos de logística.

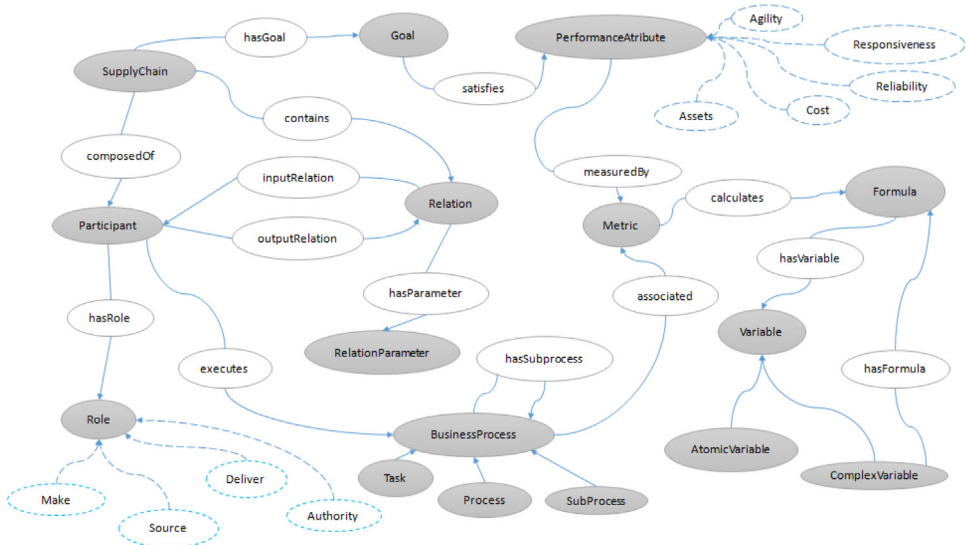


Figura 3 – Ontología SCK

Los participantes tienen procesos que deben ejecutar para cumplir con sus actividades dentro de la cadena. Esto se modela a través de la relación *executes* que relaciona

BusinessProcess con *Participant*. Se identificaron tres tipos de procesos de negocios definidos como subclases del concepto *BusinessProcess*: *Process*, *SubProcess*, y *Task* para representar los diferentes niveles de abstracción. *Process* define alcance, tipo y objetivos de rendimientos. *SubProcess* establece estrategias operativas y capacidades de un proceso. Por último *Task* define una actividad atómica dentro de un proceso. La relación *hasSubprocess* define la composición de un proceso de negocio (*BusinessProcess*) en sí mismo. Toda CS tiene un objetivo que debe ser alcanzado por la colaboración de todos sus miembros. Este concepto se modela a través de *Goal* en la ontología. Estas metas generalmente se expresan en lenguaje natural y están relacionadas a atributos de rendimiento (*PerformanceAttribute*) que pueden ser calculados a través de métricas (*Metric*). También se define la relación *associated* entre *BusinessProcess* y *Metric* indicando que un proceso de negocio tiene una métrica asociada para evaluar el grado de satisfacción del objetivo de la CS. Con el fin de medir un atributo de rendimiento, las métricas asociadas deben ser calculadas con el uso de una fórmula. Se define en la ontología la relación *calculates* que relaciona la métrica con su fórmula asociada. A su vez, una fórmula está compuesta por variables las cuales pueden ser atómicas o compuestas, estas últimas tienen asociadas otras fórmulas. En la ontología estos conceptos se modelan con la relación *hasVariable* que vincula una fórmula con sus variables y la relación de herencia entre el concepto *Variable* y los tipos de variables definidos: *AtomicVariable* y *ComplexVariable*.

3.2. Ontología HLAFed

La Figura 4 muestra la ontología *HLAFed* que contiene los términos necesarios para conceptualizar una federación HLA. Los conceptos *Federation* y *Federate* corresponden a federación y federado respectivamente. Una federación estará formada por un conjunto de federados, la relación *hasMember* entre *Federation* y *Federate* representa esta asociación. El concepto *Administrator* modela al federado encargado de administrar la federación y, como tal, es el responsable de iniciar la misma e invitar a los federados participantes a unirse a ella. Se modela la relación *iniciates* entre *Administrator* y *Federation* para representar este hecho y además la relación *hasFunction* entre *Federate* y *Administrator* para indicar que un federado cumple el rol de ser administrador de la federación. El concepto *Federation* representa la SD y por lo tanto necesita tener asociado una meta, así en la ontología se define el concepto *Objective* para representar el objetivo de la federación, y la relación *hasObjective* que relaciona *Federation* con *Objective*, indica que esa federación persigue el objetivo. Como se explicó anteriormente, los federados y federaciones tienen sus modelos de objetos que son utilizados para poder intercambiar datos e interacciones durante la simulación, constituyendo la base fundamental del éxito de la misma. El concepto *ObjectModel* representa al modelo de objeto, el mismo tiene dos sub conceptos: *FOM* y *SOM* que representan el modelo de objetos asociado a la federación y el modelo de objetos asociado a cada federado respectivamente. De esta forma las relaciones *hasFOM* y *hasSOM* relacionan *FOM* con *Federation* y *SOM* con *Federate* respectivamente. *ObjectModel* abstrae la estructura que debe tener tanto el *SOM* como el *FOM*. Para ello se identifican los conceptos que lo conforman. El concepto *POC* representa la información de contacto del responsable de la federación. La relación *contactInformation* relaciona *POC* con *ObjectModel*.

Los conceptos *Interaction* y *ObjectClass* son parte del modelo de objetos, lo cual se representa con las relaciones *hasObject* y *hasInteraction* que relacionan *ObjectModel* con *ObjectClass* e *Interaction* respectivamente. A su vez *ObjectClass* es un elemento compuesto que puede estar formado recursivamente por otro *ObjectClass* lo cual queda representado con la relación *composedOfObject* sobre sí mismo. A su vez, *ObjectClass* puede tener asociado atributos (concepto *Attribute*), los cuales a su vez tienen propiedades (concepto *Property*). Las interacciones también se pueden descomponer en otras interacciones formando una estructura jerárquica. Esta composición se expresa en la ontología a través de la relación *composeOfInteraction* que relaciona *Intaraction* con sí mismo. Las interacciones tienen parámetros y propiedades, lo cual se identifica con los conceptos *Parameter* y *Property* respectivamente. Por otro lado, el modelo de objetos contiene información de administración correspondiente a servicios prestados por el RTI durante una simulación. Estos conceptos son: *DataType*, *Time*, *Transportation*, *MOMInteraction*, *MOMObject*, *Dimension* y *Note*.

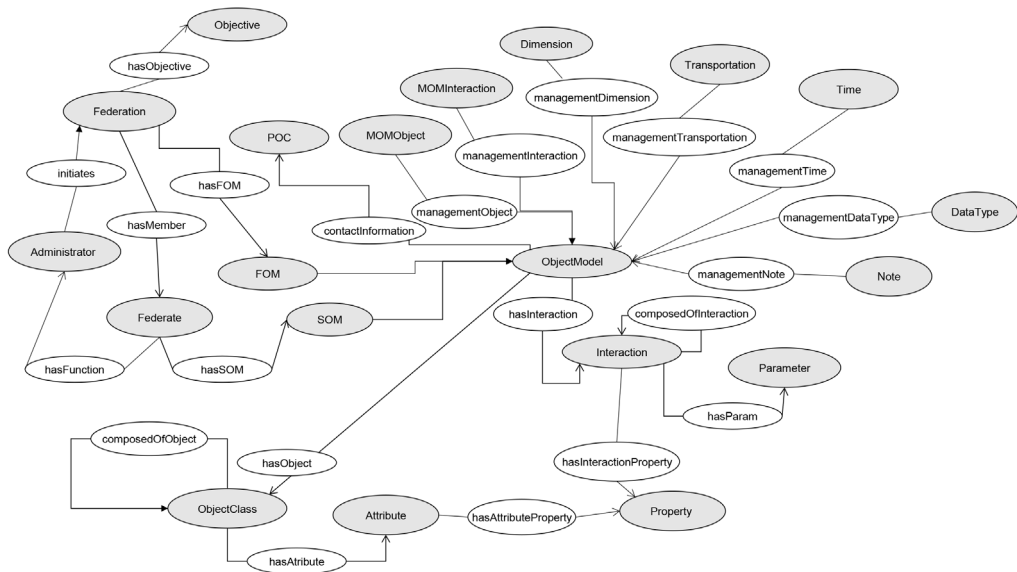


Figura 4 – Ontología HLA Fed

DataType representa las características de los tipos de datos que van a ser usados tanto por las interacciones como por los objetos definidos por el usuario. El concepto *Note* agrega información adicional al modelo de objetos que facilita el uso de los datos específicos. *Transportation* define diferentes mecanismos para transportar los datos entre federados. *Time* conceptualiza las diferentes estrategias de administración del tiempo de cada federado. *Dimension* corresponde a la organización de un grupo de objetos en dimensiones. *MOMObject* representan grupos de objetos y sus atributos

usados para describir el estado de la federación y de los federados. Finalmente, *MOMInteraction* representa el conjunto de servicios usados por un federado para adaptar, requerir, reportar o invocar servicios en nombre de otro federado. Para cada uno de estos conceptos se definen las relaciones necesarias que asocian estos conceptos con *ObjectModel*, indicando que todos ellos están formando parte de dicho modelo.

A partir de la instanciación de la ontología *SCK*, es posible derivar instancias y valores de atributos en la ontología *HLAFed* que representan los mismos conceptos pero en dominios diferentes. Así por ejemplo, el concepto de CS tiene su equivalente en federación, indicando que una federación simula el comportamiento de una CS. Además, si se define un participante como miembro de la CS, entonces habrá en la ontología *HLAFed* un federado que represente a ese participante y que forme parte de la federación asociada al concepto de CS que se define. La Tabla 1 muestra algunas de estas reglas.

#	Descripción	Regla SWRL
1	Si hay una instancia de cadena de suministro entonces habrá una instancia de federación con su mismo nombre	$\text{SupplyChain}(?s, ?name) \rightarrow \text{Federation}(?fn, ?name) \wedge \text{performs}(?fn, ?s)$
2	Si hay una instancia de <i>Participant</i> , relacionada con una <i>SupplyChain</i> a través de la relación <i>composeOf</i> , entonces existirá una instancia de un <i>Federate</i> con el mismo nombre que el participante	$\text{Participant}(?p, ?nameP) \wedge \text{SupplyChain}(?s, ?nameS) \wedge \text{composeOf}(?s, ?p) \wedge \text{Federation}(?fn, ?nameF) \wedge \text{performs}(?fn, ?s) \rightarrow \text{Federate}(?f, ?nameP) \wedge \text{hasMember}(?fn, ?f) \wedge \text{representA}(?f, ?p)$
3	Si entre dos participantes se define una relación, entonces existirá una interacción en el modelo de objetos con el mismo nombre que la relación definida	$\text{Participant}(?a) \wedge \text{Participant}(?b) \wedge \text{Relation}(?r, ?nameR) \wedge \text{outputRelation}(?a, ?r) \wedge \text{inputRelation}(?b, ?r) \wedge \text{Federation}(?fn) \wedge \text{FOM}(?fom) \wedge \text{hasFOM}(?fn, ?fom) \rightarrow \text{Interaction}(?ic, ?nameR) \wedge \text{hasInteraction}(?fom, ?ic)$
4	Si se definen parámetros para una relación entre dos participantes, entonces la interacción asociada a la relación tendrá también dichos parámetros con igual nombre y tipo de datos.	$\text{Relation}(?r, ?nameR) \wedge \text{RelationParameter}(?rp, ?nameRP) \wedge \text{RelationParameter}(?rp, ?datatype) \wedge \text{hasParameter}(?r, ?rp) \rightarrow \text{Interaction}(?ic, ?nameR) \wedge \text{Parameter}(?p, ?nameRP) \wedge \text{Parameter}(?p, ?datatype) \wedge \text{hasParam}(?ic, ?p)$
5	Si un participante tiene asociado el rol de autoridad, entonces el federado asociado a dicho participante será el administrador de la federación.	$\text{SupplyChain}(?sc) \wedge \text{Participant}(?p) \wedge \text{composeOf}(?sc, ?p) \wedge \text{Authority}(?a) \wedge \text{hasRole}(?p, ?a) \rightarrow \text{Federation}(?fn) \wedge \text{Federate}(?f) \wedge \text{hasMember}(?fn, ?f) \wedge \text{Administrator}(?ad) \wedge \text{hasFunction}(?f, ?ad) \wedge \text{initiates}(?ad, ?fn)$

Tabla 1 – Reglas de Derivación

De esta manera a medida que se crean las instancias en la ontología *SCK*, por derivación se van generando valores de instancias y atributos en la ontología que representa el aspecto dinámico de la CS, la federación. Entonces, un modelador que conoce aspectos de CS, pero no tiene conocimiento sobre SD puede ir generando la misma a partir de conceptos conocidos de CS de una manera simple.

4. Trabajos Relacionados

La SD se utiliza como una herramienta para evaluar el rendimiento de distintos escenarios de operación de una CS. Modificar un escenario en una federación implica efectuar modificaciones sobre el FOM para que refleje el comportamiento del nuevo escenario. En la mayoría de los casos, estos cambios se realizan de forma manual a través de editores XML (W3C, 2015a) o editores OMT (IEEE, 2010c) los que son propensos a introducir errores humanos de interpretación. Además, estos editores no cuentan con los mecanismos para garantizar la integridad (contener todas las etiquetas requeridas) ni la consistencia (reflejar efectivamente el comportamiento de la federación a simular) del documento. A continuación, se presentan algunas investigaciones que brindan mecanismos para evitar este tipo de errores. Las mismas se han obtenido por medio de una búsqueda en la web de las palabras clave interoperabilidad y simulación distribuida. En (Jain et al., 2015) se presenta la generación del FOM a partir de las características que posee cada federado que integra la federación. En este trabajo los autores plantean reutilizar federados existentes. Así, utilizando el lenguaje SysML (Delligatti, 2013) se describen las características que cada uno va a consumir o producir en la federación. Luego, se definen relaciones entre las características a fin de determinar el nivel de semejanza entre las mismas. Finalmente, se utiliza un algoritmo para determinar cuáles son las características más parecidas, y éstas conforman el FOM. Si bien este enfoque construye el FOM, cuenta con un alto grado de intervención humana, no presenta mecanismos para garantizar la integridad, como así tampoco para verificar los diagramas modelados en SysML.

En (Jian, Yang, & ZiYang, 2017) se presenta una solución para la generación del FOM en el contexto de CS. En este caso, el FOM presenta tres interacciones que se relacionan con cuatro roles mediante las acciones de publicar/suscribir. Las interacciones como los roles definidos por los autores son tomados del modelo SCOR. Este enfoque permite construir de manera rápida una estructura de CS basada en HLA, pero el modelado se limita a tres interacciones predefinidas sin la posibilidad de representar los objetos que comparten los federados. Además, cada federado se vincula a una interacción mediante la acción predefinida a su rol, lo que imposibilita el uso de otras acciones o interacciones. Debido a estas limitaciones, resulta imposible verificar la integridad del FOM generado.

En (Zhu & Liu, 2014) se propone la generación del FOM a través de una ontología de dominio que conceptualiza componentes de simulación de federados, con la información que pueden brindar o necesitar. Para construir la federación, se fusionan las ontologías de dominio de los distintos federados que conforman la federación, en una ontología colaborativa. De este modo, se obtiene la información relativa a la federación. Esta fusión se hace empleando una función de semejanza definida por los autores. Una vez obtenida la ontología colaborativa, se generan ontologías de tareas para determinar quién publica y suscribe una interacción, como así también, los valores para los parámetros, atributos y propiedades de cada interacción u objeto. A partir de esta ontología se puede generar automáticamente el FOM con sus respectivos valores. Tanto las ontologías de dominio como la de tareas son creadas por los usuarios. Dicha tarea requiere el conocimiento necesario tanto de simulación como de ontologías.

En (Sun, Fan, Shen, & Xiao, 2012) se presenta un método de construcción de un modelo de interoperabilidad basado en ontologías el que permite el reúso de subsistemas en

varios contextos colaborativos de desarrollo. Se enfoca principalmente en el desarrollo colaborativo de productos. Se utiliza una ontología general (denominada meta ontología) para representar conceptos de HLA, del dominio de desarrollo colaborativo de productos y de las reglas de negocio. Dicha ontología se instancia con los datos del SOM de cada federado, del escenario a simular, por medio de un proceso automático (basado en un autómata de estado finito) definido por los autores. Finalmente, se chequea la consistencia de la ontología para verificar que no existan redundancias, sinónimos y posiciones incorrectas en las categorías. Esta ontología se denomina aplicable al escenario a simular, y es utilizada para la interoperabilidad entre los federados. Este enfoque apuesta a generar ontologías aplicables, para luego fusionarlas en una gran ontología colaborativa, que abarque la mayor cantidad de escenarios de desarrollo colaborativo de productos. Como desventaja se puede remarcar que no se presenta cómo utilizar la información de las ontologías aplicables, en la construcción del FOM, a pesar de remarcarse que para el reúso de federados existentes, se deben modificar las interfaces para que cumplen con lo definido en el FOM. Esta tarea disminuye la eficiencia del reúso, es demandante de tiempo y esfuerzo.

La Tabla 2 resume las diferencias entre el enfoque SCFHLA y las propuestas analizadas en relación a indicadores que fueron considerados prioritarios en este trabajo. En la Tabla 2, para la propiedad de interoperabilidad el número 1 representa el nivel sintáctico, el 2 el nivel semántico y el 3 nivel pragmático.

#	Propiedad	SCFHLA	Jian 2017	Sun 2012	Jain 2015	Zhu 2014	XML/OMT
1	Vocabulario Comun	Si	Si	Si	Si	Si	Si
2	Intervención Humana	Baja	Media	Media	Alta	Alta	Alta
3	Garantiza Integridad	Si	No	Si	No	No	No
4	Garantiza Consistencia	Si	Si	No	Si	Si	No
5	Interoperabilidad	2	3	2	2	2	1

Tabla 2 – Comparación entre SCFHLA y otros modelos de interoperabilidad

5. Caso de Estudio

Para ejemplificar el uso de la red SCFHLA se presenta una CS llamada *Chair Co* dedicada a la producción y venta de sillas de caño. Esta cadena está formada por una fábrica, tres proveedores y tres clientes mayoristas. Los tres proveedores corresponden al proveedor de asientos, de caños y de pinturas. Los mismos reciben órdenes de suministro y entregan el producto a la fábrica. La fábrica, llamada *Sillaforte*, se dedica a fabricar sillas y venderlas a los mayoristas. Recibe los materiales desde sus proveedores, fabrica y vende a los mayoristas, de los cuales recibe las órdenes de compras. Los mayoristas son tres mueblerías que están distribuidas geográficamente, uno en Buenos Aires, otro en Mendoza y otro en Tucumán. La forma de trabajo de los mayoristas es el de mantener un determinado stock. La Figura 5 muestra la CS definida.

Usando la ontología SCK es posible obtener un modelo de la CS y definir en ella las distintas interacciones entre los participantes, tanto de materiales como de información. La construcción de este modelo conceptual permitirá obtener un modelo equivalente de simulación como se mostrará en este ejemplo.

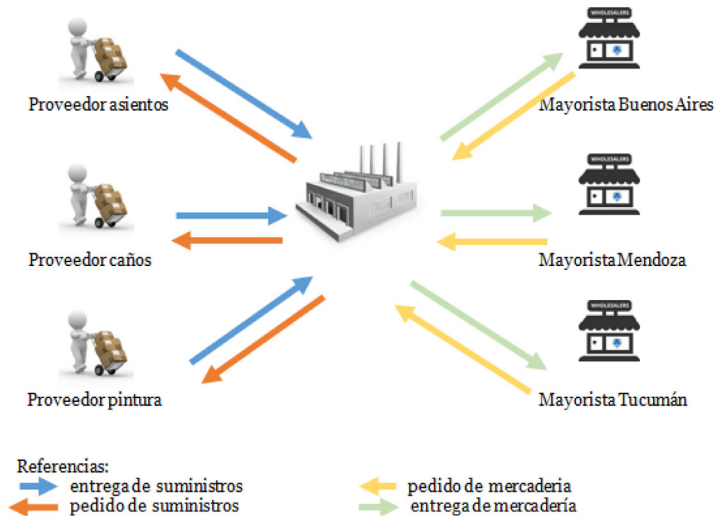


Figura 5 – Cadena de Suministro Chair Co

También es posible identificar las metas a perseguir y las fórmulas necesarias para evaluar el rendimiento. De este modo, se verifica si las metas están o no cumplidas, como también las relaciones entre los participantes. La Figura 6 muestra la instanciación de la ontología SCK representando la CS *Chair Co* como una instancia del concepto *SupplyChain* y de sus componentes, los cuales son instancias del concepto *Participant*. Se puede observar que se instanciaron los tres proveedores (P-asientos; P-pintura y P-caños), y los tres mayoristas (W-Bs.As., W-Tucumán, W-Mendoza). Todas estas instancias están relacionadas con la instancia de *SupplyChain* a través de la relación *composedOf*, que indica que forman parte de esa cadena. Una vez definidas las instancias se identifican las relaciones que hay entre ellas. Así, *Sillaforte* envía requerimientos de abastecimiento a sus proveedores y recibe de ellos el producto solicitado.

Para poder representar estas interacciones se define la relación *Rel-requerir pintura* que tiene asociado a *Sillaforte* a través de *outputRelation* y a *P-pintura* a través de *inputRelation* (*Sillaforte* à *P-Pintura*). Esto se repite para los otros proveedores. De igual forma, *P-Pintura* envía la mercadería solicitada a *Sillaforte*, que se representa con la relación *Rel-abastecer pintura* donde *Sillaforte* está asociada a través de *inputRelation*, y *P-Pintura* a través de *outputRelation* (*P-Pintura* à *Sillaforte*). Esto se repite para los otros dos proveedores. En la imagen de la Figura 7(a) sólo se presentan las relaciones con *P-Pintura*. Por otro lado, *Sillaforte* se relaciona con sus mayoristas, recibiendo solicitudes de abastecimiento y enviando el producto. En la Figura 7(a) sólo se muestra

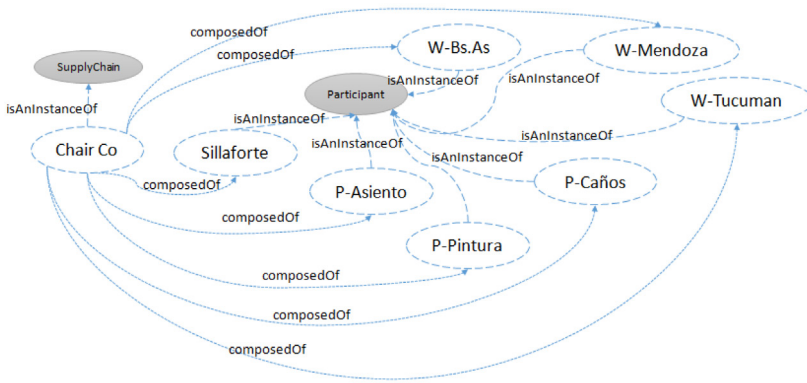


Figura 6 – Instancias de CS Chair Co y sus Participantes

las relaciones con *W-Mendoza*. Como en el caso anterior, la relación *Rel-requerir producto* tiene relacionado a *W-Mendoza* mediante *outputRelation* y a la fábrica a través de *inputRelation*; mientras que la relación de proveer mercadería (*Rel-proveer sillas*) se da a la inversa. En la Figura 7(a) se muestran los parámetros de las relaciones requerir producto y abastecer pintura.



Figura 7 (a) – Relaciones entre los Participantes

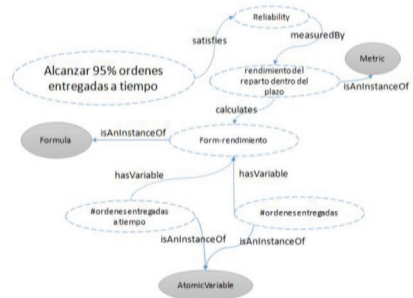


Figura 7 (b) – Definición de Métricas y Atributos de Performance

En la Figura 7(b) se muestra la definición de la meta perseguida: “alcanzar el 95% de las órdenes entregadas a tiempo”. Esta meta está asociada al atributo de performance *Reliability* y a una fórmula con dos variables: número total de órdenes entregadas y el número de órdenes entregadas a tiempo.

Una vez generado el modelo de la CS, es posible ejecutar las reglas de derivación de la Tabla 1, para generar las instancias en la ontología *HLAFed*. La Figura 8 muestra la

ejecución de las reglas 1, 2 y 3 donde se genera la instancia de la federación, las instancias de los federados (sólo se muestra para el caso de la fábrica), y la instancias de las interacciones correspondientes a las relaciones definidas entre los integrantes de la SC. Con relación a las fórmulas, usando las definidas en SCOR es posible escribir reglas en la ontología para su cálculo como muestra la siguiente regla: $\text{Formula}(\text{?p}, \text{“rendimiento del reparto dentro del plazo”}) \wedge \text{hasVariable}(\text{?p}, \text{?v1}) \wedge \text{hasVariable}(\text{?p}, \text{?v2}) \wedge \text{differentFrom}(\text{?v1}, \text{?v2}) \rightarrow \text{eval}(\text{?resultado}, (\text{v1/v2}) * 100, \text{?v1}, \text{?v2})$

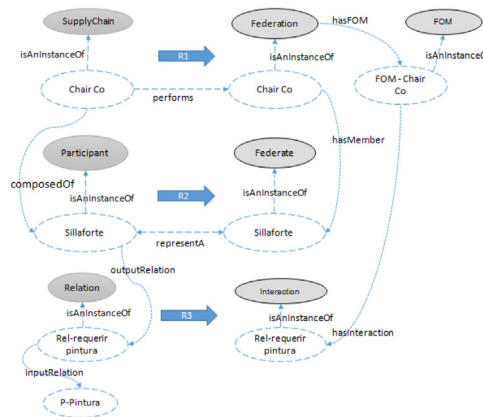


Figura 8 – Generación de Instancias en HLA Fed mediante Ejecución de Reglas de Derivación

6. Conclusiones y Trabajos Futuros

La SD tiene grandes ventajas a la hora de analizar y evaluar el funcionamiento de una CS entre ellas se mencionaron la modularidad, reúso de componentes e independencia de los miembros, siendo su principal desventaja la heterogeneidad de sus componentes. Para afrontar la necesidad de interoperabilidad entre los mismos, principalmente acordar sobre el modelo de datos a intercambiar durante una simulación, en este trabajo se presenta una red de ontologías. Por un lado, una ontología provee un vocabulario común sobre el cual acordar, definiendo la sintaxis y semántica de las interacciones entre componentes. Por otro lado, al plantear una red de ontologías, se promueve el desarrollo modular y colaborativo de las mismas facilitando su desarrollo, depuración y mantenimiento. Los conceptos modelados en la ontología SCK tienen por finalidad encontrar un vocabulario acorde para los expertos en CS que les permita estar alejados de los conceptos de simulación. De esta manera, el modelo de la cadena puede ser construido sin mayores dificultades en base a este vocabulario. Las reglas de derivación propuestas tienen la finalidad de facilitar la creación del modelo de simulación correspondiente que refleje fielmente la CS. Así una vez construido el modelo de la CS, se obtiene por ejecución de las reglas, el modelo equivalente con los conceptos de SD apropiados. Para facilitar la definición de métricas, se propone en la ontología utilizar las definidas en SCOR, como las presentadas en el ejemplo: *confianza de órdenes completadas* y *costo total de cumplimiento de una orden*. Como trabajo futuro se planea la definición de reglas que permitan asociar métricas, fórmulas y variables, a los parámetros a calcular durante la ejecución de la simulación.

Referencias

- Allocca, C., D'Aquin, M., & Motta, E. (2009). DOOR: Towards a Formalization of Ontology Relations. In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, (Vol. 1, pp. 13–20), Funchal, Portugal. <https://doi.org/10.5220/0002276400130020>
- Anagnostou, A., & Taylor, S. J. E. (2017). A distributed simulation methodological framework for OR/MS applications. *Simulation Modelling Practice and Theory*, 70, 101–119. <https://doi.org/10.1016/j.simpat.2016.10.007>
- Bébé, F., Galland, S., Gaud, N., Nicolle, C., & Koukam, A. (2014). An ontology-based metamodel for multiagent-based simulations. *Simulation Modelling Practice and Theory*, 40, 64–85. <https://doi.org/10.1016/j.simpat.2013.09.002>
- Cigolini, R., Pero, M., Rossi, T., & Sianesi, A. (2014). Linking supply chain configuration to supply chain performance: A discrete event simulation model. *Simulation Modelling Practice and Theory*, 40, 1–11. <https://doi.org/10.1016/j.simpat.2013.08.002>
- Delligatti, L. (2013). *SysML Distilled: A Brief Guide to the Systems Modeling Language*. Boston: Addison Wesley.
- Diaz, A., Motz, R., & Rohrer, E. (2011). Making Ontology Relationships Explicit in a Ontology Network. In *Proceedings of the 5th Alberto Mendelzon International Workshop on Foundations of Data Management* (p. 13). Santiago, Chile.
- Gómez-Pérez, A., Fernandez-Lopez, M., & Corcho, O. (2010). *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. First Edition*. Berlin: Springer.
- Hofmann, M., Palii, J., & Mihelcic, G. (2011). Epistemic and normative aspects of ontologies in modelling and simulation. *Journal of Simulation*, 5(3), 135–146.
- IEEE. (2010a). *IEEE standard for modeling and simulation (M & S) high level architecture (HLA) federate interface specification*.
- IEEE. (2010b). *IEEE standard for modeling and simulation (M & S) high level architecture (HLA) framework and rules*.
- IEEE. (2010c). *IEEE standard for modeling and simulation (M & S) high level architecture (HLA) object model template (OMT) specification*.
- Jain, A., Fujimoto, R., Crittenden, J., Mengmeng, L., Kim, J., & Lu, Z. (2015). Towards automating the development of federated distributed simulations for modeling sustainable urban infrastructures. In *Proceedings of the 2015 Winter Simulation Conference* (pp. 2668–2679). IEEE.
- Jian, W., Yang, H., & ZiYang, W. (2017). Research on Modeling and Simulation of Distributed Supply Chain Based on HAS. In Y. Huang, H. Wu, H. Liu, & Z. Yin (Eds.), *Intelligent Robotics and Applications* (Vol. 10464, pp. 401–412). Springer Publishing. https://doi.org/10.1007/978-3-319-65298-6_37
- Maedche, A. (2002). *Ontology Learning for the Semantic Web*. Springer US.

- Mezgár, I., & Rauschecker, U. (2014). The Challenge of Networked Enterprises for Cloud Computing Interoperability. *Computers in Industry*, 65(4), 657–674. <https://doi.org/10.1016/j.compind.2014.01.017>
- Mustafee, N., Katsaliaki, K., & Taylor, S. J. (2014). A review of literature in distributed supply chain simulation. In *Simulation Conference (WSC), 2014 Winter* (pp. 2872–2883). IEEE.
- Mustafee, N., Taylor, S. J. E., Katsaliaki, K., & Brailsford, S. (2009). Facilitating the Analysis of a UK National Blood Service Supply Chain Using Distributed Simulation. *SIMULATION*, 85(2), 113–128. <https://doi.org/10.1177/0037549708100530>
- Mustafee, N., Taylor, S., Katsaliaki, K., Dwivedi, Y., & Williams, M. (2012). Motivations and barriers in using distributed supply chain simulation. *International Transactions in Operational Research*, 19(5), 733–751. <https://doi.org/10.1111/j.1475-3995.2011.00838.x>
- Piedra, N., Chicaiza, Janneth, Quichimbo, Pricila, Saquicela, Victor, Cadme, Elizabeth, López, Jorge, & Tovar, E. (2015). Marco de Trabajo para la Integración de Recursos Digitales Basado en un Enfoque de Web Semántica. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (E3), 55–70. <https://doi.org/10.17013/risti.e3.55-70>
- Piedra, N., & Suárez, J. P. (2018). Hacia la Interoperabilidad Semántica para el Manejo Inteligente y Sostenible de Territorios de Alta Biodiversidad usando SmartLand-LD. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (26), 104–121. <https://doi.org/10.17013/risti.26.104-121>
- Pitch Technologies. (2017). Retrieved March 20, 2017, from <http://www.pitchtechnologies.com/>
- Portico. (2016). Retrieved June 30, 2016, from <http://www.porticoproject.org/comingsoon/>
- Staab, S., & Studer, R. (Eds.). (2009). *Handbook on Ontologies*. Berlin Heidelberg: Springer.
- Sun, H., Fan, W., Shen, W., & Xiao, T. (2012). Ontology-based interoperation model of collaborative product development. *Journal of Network and Computer Applications*, 35(1), 132–144. <https://doi.org/10.1016/j.jnca.2011.02.012>
- Supply Chain Council. (2012). *SCOR*. Supply Chain Council.
- Tolaba, A. C., Caliusco, M. L., & Galli, M. R. (2014). Representación del Conocimiento de la Información Geográfica siguiendo un Enfoque basado en Ontologías. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (14), 101-116. <https://doi.org/10.17013/risti.14.101-116>
- Tolk, A., Bair, L. J., & Diallo, S. Y. (2013). Supporting Network Enabled Capability by extending the Levels of Conceptual Interoperability Model to an interoperability maturity model. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 10(2), 145–160. <https://doi.org/10.1177/1548512911428457>

- Tolk, A., Diallo, S. Y., & Padilla, J. J. (2012). Semiotics, entropy, and interoperability of simulation systems—Mathematical foundations of M&S standardization. In *Simulation Conference (WSC), Proceedings of the 2012 Winter* (pp. 1–12). IEEE.
- W3C. (2004, May). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Retrieved June 27, 2017, from <https://www.w3.org/Submission/SWRL/>
- W3C. (2015a). Extensible Markup Language (XML). Retrieved April 6, 2016, from <https://www.w3.org/XML/>
- W3C. (2015b). OWL 2 Web Ontology Language Document Overview (Second Edition). Retrieved April 6, 2016, from <https://www.w3.org/TR/owl2-overview/>
- Zeigler, B. P., & Hammonds, P. E. (2007). *Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange* (1 edition). Burlington, MA: Elsevier Academic Press.
- Zhu, J., & Liu, L. (2014). A collaboration ontology modeling method for high level architecture. *Computer Modelling and New Technologies*, 18(12), 202–208.